

Systèmes de décision hybrides et apprentissage incrémental en données

Hybrid decision systems and incremental learning

Yann Prudent, Abdel Ennaji

Laboratoire PSI – FRE CNRS 2645, Université et INSA de Rouen, 76821 Mont-Saint-Aignan
{yann.prudent, abdel.ennaji}@univ-rouen.fr
<http://www.univ-rouen/psi>

Manuscrit reçu le 11 mars 2005

Résumé et mots clés

Ce papier présente un système de décision multi-classifieurs dont la conception est pilotée par la topologie des données d'apprentissage. Celle-ci est extraite grâce à l'introduction d'un nouvel algorithme d'apprentissage de carte neuronale auto-organisée qui a la propriété d'être incrémentale en données. Cette carte est utilisée en apprentissage pour distribuer la tâche de classification sur un ensemble de classifieurs. Elle permet ensuite d'activer en phase de décision le ou les classifieurs utiles pour une nouvelle donnée. De plus, le système proposé introduit un critère de confiance s'affranchissant totalement du type de classifieurs utilisés. Ce coefficient permet de contrôler plus efficacement le compromis *Erreur/Rejet*. Des résultats comparatifs sont donnés sur des exemples synthétiques, sur la base de segmentation d'images de l'UCI et sur le problème de reconnaissance de chiffres manuscrits sur des données de la base NIST.

Regroupement incremental, apprentissage distribué, carte auto-organisatrice, reconnaissance de formes.

Abstract and key words

This paper presents a multi-classifier system design controlled by the topology of the learning data. Our work also introduces a training algorithm for an incremental self-organizing map (SOM). This SOM is used to distribute classification tasks to a set of classifiers. Thus, the useful classifiers are activated when new data arrives. Comparative results are given for synthetic problems, for an image segmentation problem from the UCI repository and for a handwritten digit recognition problem.

Incremental clustering, distributed learning, self-organizing maps, pattern recognition.

1. Introduction

En reconnaissance de formes, les phases d'apprentissage et de classification constituent des étapes fondamentales qui conditionnent en grande partie les performances du système. Les techniques d'apprentissage artificiel ont connu ces dernières décennies des avancées fondamentales à travers des modèles tels que les réseaux de neurones [HAY95] ou les *Séparateurs à Vastes Marges* (SVM) [VAP95], qui montrent globalement de bonnes performances. Ces méthodes, basées sur différentes théories et méthodologies, ont été considérées dans un premier temps comme autant de solutions possibles à un même problème. Cependant, leur développement n'a pas permis de mettre en évidence la supériorité incontestable d'une méthode sur une autre pour répondre aux contraintes des applications pratiques. Il est en effet reconnu qu'au-delà des performances de reconnaissance, un système doit être capable de rejeter efficacement pour répondre aux exigences de fiabilité requises dans la plupart des applications actuelles. Ce problème de fiabilité se décline effectivement dans la possibilité de rejeter efficacement. Mais il peut être vu plus globalement comme la possibilité d'un système à fournir en plus d'une décision, une estimation de la fiabilité de celle-ci ou un coefficient de confiance qui ne soit pas simplement une probabilité *a posteriori* comme c'est souvent le cas, et qui traduit d'avantage le comportement intrinsèque du système de décision. Les conditions d'apprentissage, la gestion de conflits, la détection de zones de décisions non fiables ou encore le risque d'erreur sont autant de paramètres qui doivent être pris en compte dans l'élaboration d'un coefficient de confiance.

D'autre part, la récente émergence d'applications de fouille de données ou de textes, impliquant le traitement de grandes masses de données pour des problématiques dont la taxinomie peut être peu ou mal définie, et surtout très évolutive, limite le champ d'utilisation des outils d'analyse et d'apprentissage classiques. En effet, la plupart des méthodes reconnues efficaces dans un contexte d'apprentissage statique ne proposent aucune alternative d'évolutivité et d'adaptation dynamique pour intégrer de nouvelles connaissances, de nouvelles données ou une restructuration de problèmes déjà partiellement appris. Ce problème, bien connu actuellement en tant que dilemme stabilité/plasticité, reflète la possibilité d'un système d'apprentissage à intégrer de nouvelles données ou connaissances (plasticité) sans pour autant subir un phénomène d'oubli ou de perte des connaissances déjà acquises (stabilité).

Une solution admise dans la communauté pour résoudre ce type de problèmes consiste à adopter une conception modulaire et distribuée du système de classification [DUI02] [KUN02] [GA95]. Une telle démarche se justifie à plusieurs titres. Elle répond d'abord au principe bien connu de «diviser pour régner» qui en soit, permet de simplifier le problème d'apprentissage par l'usage d'un comité d'experts. D'autre part, l'introduction de plusieurs experts peut avoir comme conséquence une

amélioration de la fiabilité des décisions à travers la mise en place de stratégies de coopération, compétition ou de fusion. Dans ce sens, le choix des experts et la stratégie de répartition des tâches sur les différents experts sont les principaux critères de distinction des nombreuses approches proposées dans la littérature. Toutefois, cette démarche, tout en répondant partiellement au problème de l'évolutivité du système en fonction du problème grâce à sa modularité, ne peut présenter une réelle et satisfaisante réponse au dilemme stabilité/plasticité.

Dans ce contexte, ce papier tente d'apporter une contribution pour répondre à ces problèmes. Ce travail présente un système d'apprentissage multi-classifieurs dont la conception est pilotée par la topologie des données d'apprentissage. Cela suppose donc de disposer d'un outils de clustering capable d'extraire cette topologie avant de procéder à une phase de distribution de la tâche d'apprentissage sur un ensemble d'experts dédiés. Dans la perspective de disposer d'un système incrémental avec un bon compromis plasticité/stabilité, nous proposons un nouvel algorithme de clustering incrémental en données pour permettre la modélisation de la topologie des données d'apprentissage dans l'espace de représentation. Plusieurs stratégies sont ensuite possibles dans le choix des experts qui seront associés aux groupes (clusters) détectés dans l'ensemble d'apprentissage dans notre système. Ces stratégies devraient permettre de répondre à la plupart des objectifs visés (incrémentalité, fiabilité, modularité...). Actuellement, seuls des SVM sont utilisés du fait de leurs bonnes performances et des possibilités qu'ils offrent en vu d'un apprentissage incrémental. En effet, les études en cours tentent d'exploiter les propriétés des SVM pour rendre leur apprentissage incrémental en données. Plus précisément, ces travaux portent sur la possibilité de n'utiliser que les vecteurs support pour des reprises d'apprentissage sans nécessiter de disposer de l'ensemble des données ayant servis à l'apprentissage. Toutefois, toute stratégie de combinaison ou de compétition au sein de chaque cluster demeure une alternative à considérer pour plus de fiabilité et pour pouvoir adapter les traitements au contexte local.

Dans ce schéma distribué et modulaire, nos contributions concernant la fiabilité des décisions portent sur l'introduction de plusieurs niveaux de rejet dans le système. Un premier niveau de rejet est élaboré par les neurones de la carte auto-organisatrice qui pilote les classifieurs de base supervisés. Ce niveau de rejet conduit en particulier à fermer les frontières de décision des classifieurs dans l'espace de représentation. Le deuxième niveau de rejet est celui qui peut être généré par les classifieurs eux même. Il dépend donc du type de classifieurs de base choisis. Le troisième niveau de rejet est introduit grâce à un coefficient de confiance élaboré dans l'optique de prendre en compte des paramètres liés au contexte de l'apprentissage et de permettre à l'utilisateur de fixer aisément un compromis erreur/rejet adapté à la problématique considérée.

Ce papier est organisé de la manière suivante. La section 2 présente une synthèse en 2 parties successives des principales contributions respectivement dans les domaines des systèmes

multi-classifieurs et ensuite des approches de clustering par cartes auto-organisatrices. La section 3 présente notre algorithme de clustering incrémental et ensuite la stratégie adoptée dans la conception de notre système distribué multi-classifieurs et des critères de rejet introduits. La section 4 est consacrée à la validation expérimentale de notre approche sur différents jeux de données : des données synthétiques pour illustrer le comportement de nos algorithmes et ensuite sur les 2 problèmes bien connues de segmentation d'images de la base de l'UCI et en reconnaissance de chiffres manuscrits sur la base NIST. Une conclusion et des perspectives sont enfin données en section 5.

2. Systèmes Multi-Classifieurs et Cartes auto-organisées

2.1. Systèmes de classification distribués

L'utilisation d'approches d'ensembles d'experts ou encore la combinaison de classifieurs tente d'apporter une alternative aux approches classiques et monolithiques en apprentissage. Ces approches connaissent depuis plusieurs années un intérêt qui ne cesse de croître. On peut distinguer plusieurs approches dans ce domaine selon la stratégie adoptée dans la conception du système et des connaissances prises en considération. Sans être exhaustifs, on peut en effet distinguer les familles d'approches cherchant à exploiter différentes descriptions (jeux de caractéristiques) des formes qui peuvent être associées aux mêmes ou à différents classifieurs. D'autres approches s'attachent aux données d'apprentissage tels que le bagging ou le boosting soit en effectuant des rééchantillonnages des données, soit des repondérations des données après chaque apprentissage. On peut également distinguer les approches qui cherchent à combiner plusieurs classifieurs en ne faisant usage que des informations supervisées avec une multitude de règles de combinaison ou de fusion (vote majoritaire, produit, somme pondérées...) bien relatées dans la littérature [DUI02] [KUN02], de celles qui tentent d'exploiter plusieurs sources de connaissances. Cette dernière catégorie d'approches, ou approches hybrides, est celle à laquelle se réfère le plus le travail présenté dans ce papier. Ces approches hybrides s'appuient pour la plupart sur une distribution d'experts dans l'espace de description grâce à une phase préalable de catégorisation. Ainsi, Jacobs et Jordan [JJNH91] proposent une approche qui décompose l'ensemble d'apprentissage en plusieurs sous-ensembles, puis entraînent plusieurs réseaux de neurones sur ces sous-ensembles. Cette méthode fonde la distribution du problème de classification sur des informations strictement supervisées, mais ne prend pas en compte la distribution réelle des données dans l'espace de représen-

tion. Plus récemment, l'approche proposée dans [RIB98] utilise une classification ascendante hiérarchique pour déterminer les clusters dans les données d'apprentissage. Un Perceptron Multi-Couches (PMC) est ensuite associé et entraîné sur chaque cluster représentatif d'une classe détectée (au sens supervisé du terme). Les résultats obtenus avec cette approche sont équivalents à un classifieur global de type K Plus Proches Voisins (KPPV) pour des taux d'erreur maximaux, mais de bien meilleures performances quand l'erreur doit être très faible. Toutefois la classification hiérarchique ascendante, bien que très efficace sur les problèmes de *clustering* et ne nécessitant pas de fixer de paramètres *a priori* tel que le nombre de clusters, reste très coûteuse en temps et en espace.

Sans réellement appartenir aux travaux utilisant un ensemble d'experts, [GA95] et [HPG99] sont des méthodes hybrides qui utilisent le principe de spécialisation dans l'espace de description grâce à l'utilisation conjointe d'un algorithme d'apprentissage supervisé et d'un algorithme d'apprentissage non supervisé.

Dans [GA95], l'auteur commence par une phase d'apprentissage non supervisé par cartes de Kohonen [KOH82]. Cette carte permet de spécialiser des groupes de neurones d'un PMC en fonction des groupes détectés. L'utilisation d'une carte de Kohonen oblige l'utilisateur à fournir le nombre de clusters *a priori* et ne semble pas être un choix pertinent dans l'optique de l'élaboration d'un système d'apprentissage incrémental.

Dans [HPG99], l'auteur tente de palier à ces inconvénients en se servant d'un algorithme d'apprentissage non supervisé qui n'a besoin d'aucune connaissance préalable du problème : le réseau *Growing Neural Gas* [FRI95b]. Cette approche lui permet de reprendre les travaux de [GA95] sans avoir à connaître le nombre de groupes *a priori*.

L'ensemble des approches hybrides précédentes se servent donc d'un algorithme non supervisé pour répartir les tâches de classification soit sur plusieurs neurones d'un même PMC soit sur une population de PMCs experts. Il n'en demeure pas moins qu'aucune d'elles n'exploite la diversité des classifieurs proposés dans la littérature, ni ne propose une réelle alternative incrémentale aux classifieurs classiques mis à part celle introduite par la modularité du système proposé.

2.2. Cartes auto-organisatrices

Pour la conception d'un système multi-classifieurs, les méthodes hybrides présentées ci-dessus effectuent une étape préalable qui consiste à capturer la topologie des données dans l'espace de description. Ceci revient à un problème de classification automatique (catégorisation ou clustering) et implique donc de déterminer le nombre et la constitution des groupes dans l'ensemble d'apprentissage. Les techniques les plus généralement utilisées dans ce domaine sont certainement les cartes auto-organisatrices [KOH82] et les méthodes de regroupement par partitionnement [MAC67]. Les méthodes par partitionne-

ment nécessitent pour la plupart de fournir un certain nombre de paramètres *a priori* tels que le nombre ou la distribution des clusters dans les données. Ces connaissances *a priori* sont difficiles à estimer pour des problèmes réels et complexes et vont à l'encontre du principe d'évolutivité que nous nous proposons d'aborder. Notre choix s'est donc porté sur les cartes auto-organisatrices qui partagent toutes un certain nombre de propriétés. La structure des cartes auto-organisatrices est un graphe composé d'un ensemble N de nœuds (neurones) et d'un ensemble C de connexions entre ces neurones. Chaque neurone n a un vecteur de référence w_n qui correspond à sa position dans l'espace de description. L'adaptation des vecteurs de référence se fait pour chaque nouvelle donnée en modifiant la position du neurone gagnant et de ses voisins dans le graphe. Pour une donnée ξ , le neurone gagnant n_1 est :

$$n_1 = \arg \min_{n \in N} \text{dist}(\xi, w_n). \quad (1)$$

La mise à jour de la position du neurone gagnant dans l'espace de description se fait de la manière suivante :

$$\Delta w_n = \varepsilon(t) h_{n,n_1} \|\xi - w_n\| \quad (2)$$

où $\varepsilon(t)$ est le pas d'adaptation, et h_{n,n_1} une fonction de voisinage.

Une distinction peut être effectuée entre deux types de cartes auto-organisatrices :

- les cartes statiques, dont le nombre de neurones est fixe et déterminé *a priori*,
- les cartes dynamiques, ou constructives, pour lesquelles le nombre de neurones varie.

Parmi les modèles de cartes auto-organisatrices statiques, le plus connu est sans doute le modèle de Kohonen. Cette carte a une structure composée d'hyper-rectangles de dimension k fixée *a priori*. Le réseau est donc constitué de telle manière que chaque neurone appartienne à un k -hyper-rectangle. L'adaptation des vecteurs de référence se fait comme décrit dans l'équation 2 avec :

$$\varepsilon(t) h_{n,n_i} = \begin{cases} f(t) & \text{si } n \in V_{n_i}(t) \\ 0 & \text{sinon} \end{cases}$$

où $0 \leq f(t) \leq 1$ est une fonction monotone décroissante et $V_{n_i}(t) = \{n \in N / \exists \text{ un chemin entre } n_1 \text{ et } n \text{ plus petit que } R_{n_i}(t)\}$, $R_{n_i}(t)$ étant une fonction entière décroissante. La décroissance simultanée de la fonction f et du rayon de voisinage permet d'explorer grossièrement l'espace de description au début du processus d'apprentissage, puis de réaliser un affinement de la position des neurones dans la phase terminale de l'apprentissage.

Neural Gas [MS91] est un autre modèle de cartes auto-organisatrices de quantification vectorielle et qui peut être combiné à un apprentissage Hebbien compétitif [MAR93]. La combinai-

son de ces deux méthodes est appelée *Topology-representing network* [MS94]. Le principe est le suivant : pour chaque donnée, les vecteurs de référence des k plus proches neurones de cette donnée sont modifiés de manière à diminuer l'erreur engendrée. La variable k diminue au cours du temps. Le principe de l'apprentissage Hebbien compétitif est simplement de créer une connexion entre les deux neurones les plus proches de la nouvelle donnée.

Dans la catégorie des cartes dynamiques, on compte la méthode *Growing Cell Structure* (GCS) [FRI94] qui est une carte dont la structure est formée d'hyper-tétraèdres de dimension k fixée *a priori*. Un k -hyper-tétraèdre est un graphe complet de taille $k + 1$ (un segment quand $k = 1$, un triangle quand $k = 2$, ...). Le réseau est donc constitué de telle manière que chaque neurone appartienne à un k -hyper-tétraèdre. Le modèle est initialisé avec un seul k -hyper-tétraèdre. À chaque itération l'adaptation des vecteurs de référence est effectuée avec :

$$\varepsilon(t) h_{n,n_1} = \begin{cases} \varepsilon_b & \text{si } n = n_1 \\ \varepsilon_m & \text{s'il y a une connexion entre } n \text{ et } n_1 \\ 0 & \text{sinon} \end{cases}$$

De plus, à chaque fois qu'une nouvelle donnée est apprise, l'erreur quadratique qu'elle a engendrée est accumulée par le neurone gagnant n_1 .

$$\Delta E_{n_1} = \|w_{n_1} - \xi\|^2$$

Toutes les λ itérations, λ étant fixé *a priori*, le neurone q ayant le plus d'erreur accumulée est déterminé. Un nouveau neurone est inséré en coupant en deux la connexion la plus longue émanant de q . Ensuite, des connexions sont ajoutées pour préserver la structure en hyper-tétraèdres.

Le modèle *Growing Grid* (GG) [FRI95a] est basé sur un algorithme d'apprentissage identique au réseau GCS, mais sa structure est formée d'hyper-rectangles. Pour maintenir cette structure, il est donc nécessaire d'ajouter une ligne ou une colonne entière à chaque fois que le réseau doit augmenter de taille.

Le réseau *Growing Neural Gas* (GNG) [FRI95b] lui, n'impose aucune contrainte sur sa structure. Les connexions sont mises à jour par un apprentissage Hebbien Compétitif. L'adaptation des positions des neurones dans l'espace de description est identique à celle du réseau GCS. Toutes les λ itérations, λ étant fixé *a priori*, le neurone q ayant le plus d'erreur accumulée est déterminé. Un nouveau neurone est inséré entre q et le neurone connecté à q qui a accumulé le plus d'erreur. Au cours de l'apprentissage les connexions vieillissent et sont supprimées à partir d'un seuil. Si cette opération isole un neurone, celui-ci est également supprimé.

Certains des réseaux présentés permettent d'apprendre de nouvelles données de façon dynamique. Certains même, comme le GNG, n'ont pas de contraintes sur leur structure, ce qui leur permet d'avoir des dimensions variables dans différentes régions de

l'espace de description. Mais aucun des modèles présentés ne garantit la préservation des connaissances déjà acquises. De plus, tous les réseaux évolutifs présentés ajoutent un neurone toutes les λ étapes, ils peuvent donc engendrer un nombre de neurones soit trop important (sur-apprentissage), soit insuffisant (sous-apprentissage) selon le nombre de fois où chaque donnée est présentée au réseau.

3. Contributions

Dans la perspective de construire un système d'apprentissage incrémental en données, d'une fiabilité accrue par rapport aux approches classiques et dans la mesure du possible, capable de fournir à l'utilisateur le moyen de paramétrer efficacement le compromis erreur/rejet, il nous a semblé opportun de s'inspirer des approches multi-classifieurs hybrides basant la distribution du problème sur des critères topologiques des données d'apprentissage. Il est donc indispensable de disposer d'un algorithme de clustering incrémental et ne nécessitant pas de connaissances *a priori* « fortes » tel que le nombre ou la distribution des clusters. Notre choix s'est orienté vers des approches par cartes auto-organisatrices. Dans cette partie, nous décrivons donc un nouvel algorithme d'apprentissage incrémental pour une carte auto-organisatrice avant de présenter une utilisation possible de celle-ci pour la conception d'un système distribué.

3.1. IGNG : Incremental Growing Neural Gas

Le modèle que nous proposons dans ce papier est un modèle constructif et incrémental en données de cartes auto-organisatrices. Comme le modèle *Growing Neural Gas*, notre approche n'impose aucune contrainte sur la structure du réseau. Celui-ci est mis à jour de façon continue par un apprentissage Hebbien compétitif. Contrairement aux réseaux décrits précédemment, celui-ci possède deux types de neurones :

- des neurones matures,
- des neurones embryons.

Chaque neurone possède un âge, un vecteur de référence et un type (mature ou embryon). Chaque connexion possède un âge. Afin de construire notre réseau, nous effectuons avant l'apprentissage de chaque donnée ξ un test défini par l'équation 3 où σ est un seuil fixé *a priori*. Ce test sert à vérifier si l'insertion d'un neurone est nécessaire. S'il n'existe pas au moins deux neurones qui satisfont cette équation, alors l'ajout d'un nouveau neurone est effectué.

$$\text{dist}(\xi, w_n) \leq \sigma \quad (3)$$

Quand un neurone est inséré, c'est un neurone de type embryon avec un âge nul.

Initialement le réseau est vide. À chaque itération, nous cherchons le neurone gagnant n_1 (équation 1). Si celui-ci n'existe pas (réseau vide) ou s'il ne satisfait pas la condition de l'équation 3, un nouveau neurone est ajouté avec $w_{new} = \xi$. Ce premier cas est présenté par la figure 1. La Figure 1.a représente un réseau et une nouvelle donnée qui ne satisfait pas la condition de l'équation 3. La Figure 1.b montre l'adaptation de ce réseau à la nouvelle donnée par la création d'un nouveau neurone. Un deuxième cas peut se produire quand un neurone gagnant n_1 satisfaisant l'équation 3 existe. Nous recherchons alors le deuxième neurone n_2 le plus proche de la donnée. Si n_2 n'existe pas ou s'il ne satisfait pas le test, un nouveau neurone est ajouté avec $w_{new} = \xi$ et une connexion est créée entre ce neurone et n_1 . La figure 2 illustre ce cas.

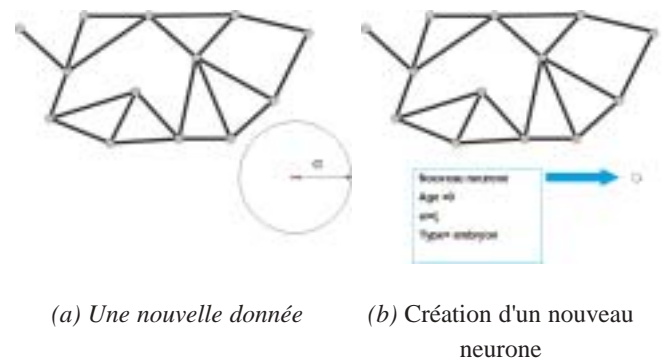


Figure 1. Insertion d'un neurone quand la donnée à apprendre est trop loin de son neurone gagnant

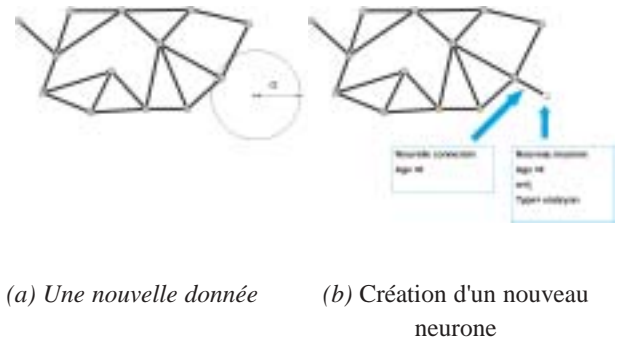


Figure 2. Insertion d'un neurone quand la donnée à apprendre est trop loin de son deuxième neurone gagnant

Enfin, le dernier cas de figure correspond à l'existence d'au moins deux neurones qui satisfont le test de l'équation 3. Aucun neurone n'est ajouté dans cette situation et l'adaptation des vecteurs de référence se fait comme pour le GCS avec dans l'équation 2 :

$$\varepsilon(t)h_{n,n_1} = \begin{cases} \varepsilon_b & \text{si } n = n_1 \\ \varepsilon_m & \text{s'il y a une connexion entre } n \text{ et } n_1 \\ 0 & \text{sinon} \end{cases}$$

L'âge des connexions émanant de n_1 est incrémenté. Comme dans l'apprentissage Hebbien compétitif, nous ajoutons une connexion entre les deux plus proches neurones de la donnée. Si celle-ci existe déjà son âge est réinitialisé à zéro. Toutes les connexions ayant un âge supérieur à a_{max} sont supprimées, et si ceci a pour conséquence l'isolement d'un neurone, celui-ci est supprimé. L'âge de tous les neurones connectés à n_1 est incrémenté, et tous les neurones embryons avec un âge supérieur à a_{mature} deviennent des neurones matures. L'apprentissage de ce réseau est synthétisé dans l'algorithme 1.

Algorithme 1 : L'algorithme *Incremental Growing Neural Gas*

Données a_{max} , a_{mature} , σ , **Base d'apprentissage** S
début

tant que le critère d'arrêt n'est pas satisfait faire

Prendre une donnée $\xi \in S$; Trouver n_1 le neurone gagnant ;

si $dist(\xi, w_{n_1}) \geq \sigma$ **alors**

insérer un nouveau neurone embryon avec $w_{new} = \xi$;

sinon

Trouver n_2 le 2ème plus proche neurone de la donnée

si $dist(\xi, w_{n_2}) \geq \sigma$ **alors**

insérer nouveau neurone embryon avec $w_{new} = \xi$;

créer une connexion entre n_1 et n_2 ;

sinon

$w_{n_1} + = \varepsilon_b(\xi - w_{n_1})$;

$w_m + = \varepsilon_m(\xi - w_m)$; (m voisins directs de n_1)

Incrémenter âge connexions émanants de n_1 ;

si n_1 et n_2 sont connectés **alors**

$age_{n_1 \rightarrow n_2} = 0$;

sinon

créer une connexion entre n_1 et n_2 ;

Supprimer connexions avec un âge supérieur à a_{max}

Si isolement d'un neurone : le supprimer

Incrémenter l'âge de tous les voisins directs de n_1 ;

pour chaque neurone embryon n **faire**

Si $age(n) \geq a_{mature}$: n devient mature

fin

Les paramètres qui entrent en jeu dans cet algorithme sont :

- a_{mature} , a_{max} ,
- ε_b , ε_m ,
- σ .

Le paramètre a_{max} correspond à l'âge maximal que peut avoir une connexion. Les paramètres ε_b et ε_m correspondent au pas d'adaptation respectivement du neurone gagnant et de ses voisins. Si l'on compare ce réseau aux cartes de Kohonen, la phase d'exploration grossière de l'espace de description correspond à l'ajout de nouveaux neurones, et l'affinement de la position des neurones dans la phase finale correspond à l'adaptation de la position des neurones. Si on reste dans l'ordre de 10^{-2} pour ε_b

et de 10^{-3} pour ε_m , ces deux paramètres n'influent que très peu. La notion de neurones matures ou embryons a surtout été introduite pour doter notre réseau de tolérance au bruit. Le paramètre a_{mature} correspond donc au nombre d'activations qu'il faut à un neurone pour que celui-ci ne soit plus considéré comme résultant de données bruitées. Par conséquent, ce paramètre est à régler en fonction de la qualité de la base d'apprentissage.

σ est le paramètre qui demande le plus de connaissances sur le problème à traiter. Il correspond au rayon de l'hyper-sphère représentant le volume maximal dans l'espace de description que doit considérer un neurone en dehors duquel un rejet est généré. Il est le paramètre le plus important dans notre approche à être encore malheureusement fixé *a priori*. Mais il représente néanmoins un moyen supplémentaire de régler le niveau de rejet du système.

Notons enfin que l'algorithme proposé ne conduit pas au phénomène de *sur-apprentissage* rencontré par les autres réseaux évolutifs, puisque l'ajout d'un nouveau neurone se fait uniquement lorsque c'est nécessaire. Cet algorithme est par ailleurs véritablement incrémental en données contrairement à la plupart des autres méthodes classiques, plus particulièrement le GNG. Cette propriété sera largement évoquée dans la section résultats. De plus, la plupart des paramètres nécessaires à son fonctionnement, identiques au GNG et constants dans le temps et au cours des différents apprentissages, ne nécessitent que peu de connaissances *a priori* pour les fixer. En effet, seul σ nécessite de telles connaissances (qui peuvent être estimées par expérimentation), mais c'est également une manière d'intégrer le contexte dans le processus d'apprentissage.

3.2. Conception d'un système distribué

La structure topologique représentée par le réseau *Incremental Growing Neural Gas* (IGNG) est utilisée pour distribuer un problème d'apprentissage supervisé sur un ensemble de classifieurs. Nous associons à chaque neurone mature n_i du réseau IGNG un sous-ensemble s_i de la base d'apprentissage S tel que :

$$s_i = \{\xi \in S / d(\xi, w_{n_i}) \leq \sigma_i\}$$

σ_i est fixé *a priori* et devrait faire l'objet d'une estimation en phase d'apprentissage du IGNG dans une version plus élaborée de notre système. Ce paramètre peut être vu de la même manière que le paramètre σ du réseau IGNG. Comme nous le montre la figure 3, chaque neurone est donc associé à une zone d'influence hyper-sphérique de rayon σ_i . Cette méthode a pour avantage de fermer les frontières de décision engendrées par les classifieurs (cf. fig. 4). En effet, la génération de frontières de décision ouvertes est un des principaux défauts de beaucoup de classifieurs tels que les PMCs [GS98]. Ainsi, bien que reconnus comme performants, ils ne peuvent rejeter de manière efficace les données n'appartenant à aucune classe. De la même manière

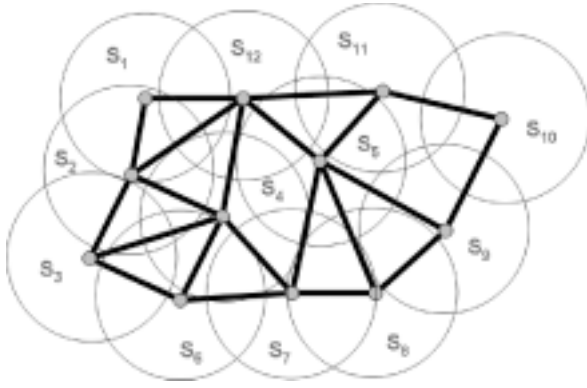
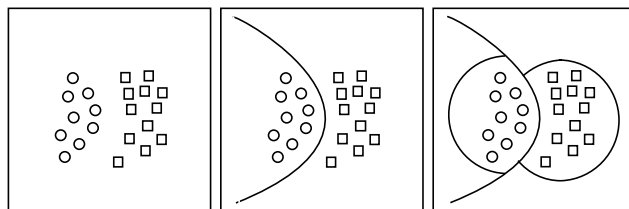


Figure 3. Distribution de l'ensemble d'apprentissage S en plusieurs sous-ensembles s_i .



(a) Données d'apprentissage (b) Frontières générées par un SVM (c) Frontières générées par notre approche

Figure 4. Fermeture des frontières de décision.

re, Les SVMs, malgré leurs capacités de modélisation et de généralisation [TFCJW03, TFCJW04] restent difficiles à paramétrer convenablement pour un comportement en rejet efficace. Dans notre système, un premier niveau de rejet est introduit à travers le paramètre σ_i . En effet l'union de tous les sous-ensembles s_i n'est pas forcément égale à l'ensemble d'apprentissage. Les données n'appartenant pas à cette union ne sont pas apprises et sont rejetées par notre système (en apprentissage comme en décision).

Une fois la base d'apprentissage décomposée en plusieurs sous-ensembles, notre système a la possibilité d'entraîner plusieurs classifieurs de différents types (PMC, SVM, KPPV, ...), ou aux paramètres d'apprentissage différents (noyaux, nombre de neurones, ...) pour chacun de ces sous-ensembles. Plusieurs stratégies peuvent ensuite être implantées à ce stade. On peut en effet soit sélectionner le ou les classifieurs adéquats au problème, soit s'orienter vers des techniques de combinaison par des méthodes proposées dans la littérature [GR00] [RG02] [KUN02].

Nous appelons par la suite l'association «neurone, classifieurs, sous-ensemble d'apprentissage» un GNeurone. La décision d'un GNeurone, décision prise localement, résulte donc de la fonction de combinaison des classifieurs associés à celui-ci. Pour les raisons évoquées en introduction, seuls des SVMs sont utilisés à ce stade de l'étude. Ainsi, pour chaque GNeurone, son sous-

ensemble d'apprentissage est analysé pour identifier les classes (au sens supervisé) qui le composent et un ensemble de SVMs binaires sont alors entraînés sur ce sous ensemble d'apprentissage. L'utilisation des SVMs (méthode non encore incrémental en données), limite forcément le caractère incrémental du système global. Cette limitation pourrait être contournée par la mise au point de techniques supervisées incrémentales [LESG95, HPG99, LW03] ou encore procéder à la supervision de la carte auto-organisée (dans le même sens que LVQ), mais au détriment des bonnes capacités des SVMs. Une technique très simple de supervision du IGNG est par ailleurs utilisée dans la section Expérimentations pour la validation de notre approche. Cette dernière technique de supervision du IGNG est par ailleurs utilisée dans la section Expérimentations pour la validation de notre approche.

Lorsqu'une nouvelle donnée ξ à classifier est présentée au système, tous les GNeurones répondants à la condition $d(\xi - w_i) \leq \sigma_i$ sont sélectionnés pour la décision. Nous choisissons alors au plus les K plus proches GNeurones de la nouvelle donnée dans l'espace de description (K est fixé *a priori* dans le même sens qu'un classifieur type KPPV). Dans la suite, notre système sera présenté sous la dénomination KPPG pour K Plus Proches GNeurones.

Une fois les K GNeurones sélectionnés, les décisions de chacun sont pondérées par un coefficient de confiance qui tient compte de deux facteurs, importants à notre sens, pour la fiabilité d'une décision :

- le nombre de données d'apprentissage du GNeurone ($|s_i|$),
- la distance entre celui-ci et la donnée à classifier.

Ce coefficient de pondération v associé à chaque GNeurone sélectionné est défini par :

$$v = \frac{|s_i|}{|s_i| + \beta d(\xi, w_{n_i})}$$

β est une constante permettant de définir un compromis entre la distance entre la nouvelle donnée et le GNeurone, et la taille du sous-ensemble d'apprentissage de celui-ci. On peut donc générer une liste de décisions possibles associées chacune à un coefficient de confiance. Dans le cas où l'on veut obtenir un taux de reconnaissance maximal sans contraintes particulières relatives au contexte, la décision retenue est celle avec la confiance maximale.

4. Validations expérimentales

La validation de nos approches aussi bien concernant l'algorithme de clustering uniquement, que son utilisation pour la conception d'un système multi-classifieurs a été effectuée à travers plusieurs expérimentations. Nous avons utilisé deux bases

synthétiques en petite dimension pour comparer notre approche IGNG au réseau GNG pour la visualisation des données et la modélisation de la topologie. La première base est une distribution de données proposée par Martinetz et Schulten [MS91] pour tester les capacités du modèle *Neural Gas*. Cette distribution a été réutilisée ensuite par Fritzke [FRI95b] pour montrer que le réseau GNG peut apprendre d'importantes relations topologiques en formant une structure avec des dimensions différentes. La deuxième distribution est un problème « jouet » que nous proposons pour montrer l'apport du point de vue incrémental de notre réseau pour la description de la topologie. Nous avons également procédé à la validation de nos contributions sur deux problèmes réels pour mettre en évidence l'apport de nos approches en généralisation sur des problèmes réels en dimensions élevées et indépendamment de la pertinence des caractéristiques utilisées. Le premier problème est issu de la base de segmentation d'images de l'UCI [MA92]. Ce problème est composé d'une base d'apprentissage de 210 exemples et d'une base de test de 2100 exemples. L'espace de description est de dimension 20 et le problème composé de 7 classes (BRICKFACE, SKY, FOLIAGE, CEMENT, WINDOW, PATH, GRASS). Le deuxième problème concerne le domaine de la reconnaissance de chiffres manuscrits sur la base NIST. Il a été abordé avec deux espaces de description différents. Le premier espace de caractéristiques considéré est constitué par les 85 caractéristiques (niveaux de gris) issues d'une pyramide de résolution à 4 niveaux (1+4+16+64) [BB82].

La figure 5(a) donne un exemple des chiffres de la base de données NIST et la figure 5(b) illustre la représentation retenue sur l'exemple d'un chiffre 2. Les bases d'apprentissage et de test sont composées respectivement de 2626 et de 2621 exemples. Le deuxième espace de représentation retenu pour ce même problème est de dimension 33 et se base sur les descripteurs de Fourier-Mellin [AOC⁺00]. Cette base est composée de 10000 exemples pour l'apprentissage et 60000 pour le test. Dans toutes les expérimentations effectuées, nous avons fixé pour le GNG et le IGNG $a_{max} = 100$.

4.1. IGNG vs GNG en apprentissage passif

Dans cette section nous comparons notre réseau IGNG au réseau GNG sur l'aspect de l'apprentissage topologique (clustering) passif, mais également sur les performances en classification. Nous appelons apprentissage passif, un apprentissage effectué en une fois avec toutes les données disponibles.

• Distribution de Martinetz

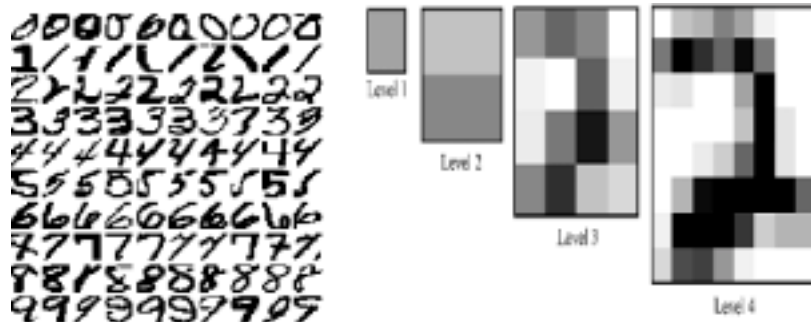
La distribution de la figure 6 représente une distribution de données pour illustrer une topologie complexe à modéliser (les points d'apprentissage ne sont pas représentés, seul le résultat de l'apprentissage est illustré). Elle a été proposée par Martinetz et Schulten [MS91] pour tester les capacités du modèle *Neural Gas*, et réutilisée par Fritzke pour le GNG. Cet exemple montre le résultat obtenu avec notre modèle, résultat équivalent au GNG et aux autres méthodes pour la visualisation de la topologie de données dans le cadre d'un apprentissage passif.

• Reconnaissance de chiffres manuscrits

Cette expérimentation, menée sur la base NIST en dimension 85, a pour but de comparer les performances en classification de notre réseau avec celles du réseau GNG. Les cartes auto-organisatrices ont été supervisées en associant à chaque neurone la classe issue du vote majoritaire sur les étiquettes des données qui l'ont activé pendant l'entraînement. Le nombre de cycles d'apprentissage d'un réseau correspond au nombre de fois où toute la base d'apprentissage a été présentée. Le tableau 1 résume les résultats obtenus pour cette expérimentation. Ce tableau nous montre que le réseau IGNG obtient des résultats équivalents en classification avec beaucoup moins de cycles d'apprentissage et moins de neurones (313 neurones matures et 9 embryons qui ne sont pas utilisés en classification).

4.2. IGNG vs GNG en apprentissage incrémental

Par opposition à l'apprentissage passif, nous rappelons que par apprentissage incrémental, nous entendons la possibilité de



(a) Chiffres de la base NIST

(b) Pyramide de résolution à 4 niveaux (chiffres 2)

Figure 5. Exemples de chiffres et représentation retenue.

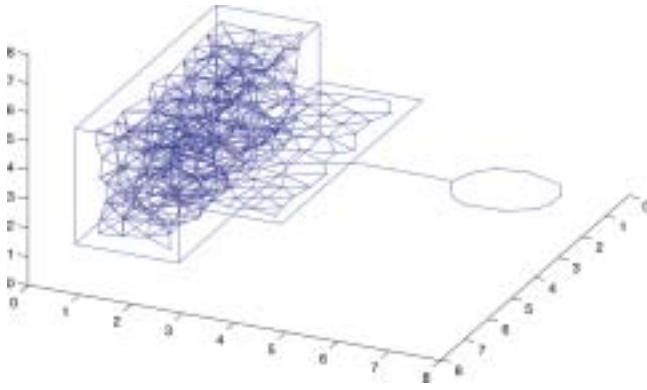


Figure 6. Réseau IGNG pour la visualisation de données sur le problème de Martinetz.

Tableau 1. Résultats en généralisation obtenus sur la base de chiffres manuscrits dans le cas d'un apprentissage passif.

carte	cycles	λ	ε_b	ε_m	σ	nb de neurones	reconnaissance
GNG	50	400	0.1	0.006	–	330+0	91.44 %
IGNG	10	–	0.01	0.002	2.7	313+9	91.71 %

reprise d'apprentissage uniquement avec les nouvelles données et sans avoir à disposition les données déjà apprises par le réseau.

• Problème synthétique

Nous proposons ici un problème synthétique [PE04] composé de deux classes en dimension 2 avec une distribution sphérique pour la première classe et cubique pour la deuxième tel que illustré sur la Figure 7. L'apprentissage incrémental est effectué en deux étapes, une première phase où le réseau apprend la topologie de la première classe uniquement, puis dans un deuxième temps, les données de la deuxième classe sont présentées au réseau. Ce problème a pour but de montrer les limites du réseau GNG pour la visualisation de la topologie dans le contexte d'un

apprentissage incrémental. La figure 7.a nous montre que pour pouvoir apprendre la topologie de la deuxième classe (distribution cubique), le réseau GNG subit une détérioration des connaissances acquises sur la topologie de la première classe (distribution sphérique). Contrairement au réseau GNG, notre réseau se comporte parfaitement et préserve bien les connaissances déjà modélisées comme illustré sur la figure 7.b.

• Reconnaissance de chiffres manuscrits

Pour cette expérimentation réalisée sur la base de chiffres manuscrits en dimension 85, nous avons découpé la base d'apprentissage en quatre parties (525 exemples chacune). L'apprentissage des deux modèles s'est fait par présentations successives des quatre parties de la base. Ici aussi, le réseau est supervisé en utilisant un vote majoritaire sur les étiquettes des données pour chaque neurone pendant l'entraînement. Le tableau 2 résume les résultats obtenus.

Tableau 2. Résultats obtenus en reconnaissance de chiffres manuscrits pour un apprentissage incrémental.

carte	cycles	λ	ε_b	ε_m	σ	nb de neurones	reconnaissance
GNG	50	400	0.1	0.006	–	328+0	81.29 %
IGNG	10	–	0.01	0.002	2.7	244+16	90.18 %

Ces résultats confirment sur un problème réel les observations précédentes. Notre réseau se prête bien à un apprentissage incrémental puisque son comportement reste stable dans ces conditions, contrairement au GNG dont les performances se dégradent de manière importante. En effet, la baisse de performances des deux modèles par rapport aux résultats obtenus en apprentissage passif sont de l'ordre de 10 % pour le GNG alors qu'elle est de l'ordre de 1 % pour notre modèle. Notre approche répond donc bien au dilemme «plasticité/stabilité» introduit auparavant.

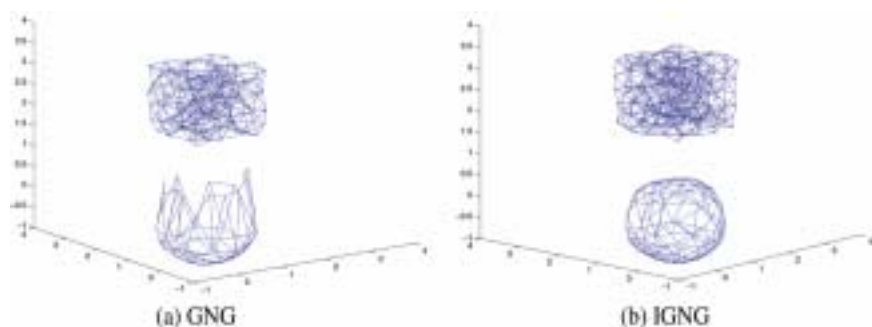


Figure 7. Comportement du GNG (a) et du IGNG (b) dans le cas d'un apprentissage incrémental.

4.3. Conception d'un classifieur distribué

Malgré les possibilités de combinaison et de panachage de classifieurs supervisés variés au sein de chaque neurone de la carte, seuls des SVMs ont été utilisés dans cette version préliminaire du travail. Ce choix se justifie par les bonnes capacités des SVM, mais également dans la mesure où les résultats obtenus dans ces conditions ne peuvent que être améliorés par une telle stratégie de diversification. De la même manière, le coefficient de confiance n'a pas encore fait l'objet de validation expérimentale à l'heure actuelle, mais sa présentation nous semble fondamentale dans le contexte de ce travail. Il n'en demeure pas moins que ces extensions sont une des perspectives à très court terme du travail.

Nous avons procédé à la comparaison en terme de performances de notre système à des classifieurs bien connus et éprouvés, le KPPV (K Plus Proches Voisins) et les SVMs binaires. L'objectif d'une telle démarche est de procéder à la validation de notre approche relativement à des approches classiques indépendamment des conditions particulières des applications considérées. En effet, les choix faits en terme de taille des bases d'apprentissage, de test et de jeux de caractéristiques sont motivés par la volonté de mettre en avant les capacités de généralisation des diverses techniques malgré les phénomènes de surapprentissage qui peuvent se produire dans certaines situations. Le noyau utilisé pour les SVMs a été déterminé par un procédé d'essai/erreur en utilisant les noyaux suivants :

- noyau linéaire: $K(x,y) = x * y$,
 - noyau polynomial: $K(x,y) = (\gamma xy + c_0)^{degree}$,
 - noyau RBF: $K(x,y) = \exp(-\gamma |x - y|^2)$,
 - noyau sigmoid: $K(x,y) = \tanh(\gamma xy + c_0)$,
- en faisant varier γ entre 0 et 1 par pas de 0.1, et *degree* entre 2 et 5 par pas de 1, nous avons pris $c_0 = 0$.

• Segmentation d'images de l'UCI

Le meilleur noyau SVM obtenu pour ce problème est un noyau linéaire. Nous avons pris comme paramètres pour notre système $\sigma_i = 250 \forall i$ et $k = 1$ (respectivement l'étendue de l'hyper-sphère d'influence de chaque GNeurone et le nombre de GNeurones activés à chaque décision). Le tableau 3 présente les résultats obtenus.

Le tableau 3 montre clairement sur ce problème bien connu, des résultats équivalents de notre système et des SVMs, et bien

meilleurs que le KPPV. Et ce, malgré une taille très réduite de la base d'apprentissage par rapport à la base de test.

• Reconnaissance de chiffres manuscrits

Le noyau SVM utilisé pour ce problème est un noyau RBF avec $\gamma = 0.2$.

Le tableau 3 présente les résultats obtenus pour l'espace de description en dimension 33 et le tableau 5 l'espace de description en dimension 85.

Les résultats du tableau 4 confirment les tendances observées sur le problème de segmentation d'images de l'UCI, pour ce problème dont le nombre de classes, la dimension de l'espace de représentation et la taille des bases sont plus importants. Les résultats du tableau 5 et les courbes de la figure 8 représentent l'évolution du taux de reconnaissance en fonction du taux d'erreur fixé. Ces résultats ont été obtenus pour chaque classifieur considéré par un vote unanime des K plus proches voisins et K GNeurones respectivement pour le KPPV et le KPPG en augmentant K jusqu'à ce que le taux d'erreur souhaité soit respecté (le taux maximum d'identification est obtenu quand $k = 1$).

Tableau 4. Taux de reconnaissance et d'erreur des trois classifieurs sur NIST(33)

classifieur	reco (%)	erreur (%)
KPPV	85.72	14.28
SVM	90.15	9.85
KPPG	90.79	9.21

Comme les SVMs ne sont pas, à notre connaissance, capables de rejeter efficacement, il est impossible de paramétrer un tel comportement et obtenir une courbe pour ce classifieur. Ces résultats montrent des performances au moins équivalentes aux SVMs et meilleures que le KPPV pour des taux d'erreurs non paramétrables. Toutefois, contrôler efficacement le taux d'erreur est devenu une exigence incontournable dans la plupart des applications de reconnaissance de formes actuellement. Et dans ce contexte, notre approche offre, en plus d'être incrémentale en données, des performances bien meilleures.

Tableau 3. Taux de reconnaissance et d'erreur des trois classifieurs sur la base Image de l'UCI.

classifieur	reco (%)	erreur (%)
KPPV	87.57	12.43
SVM	93.95	6.05
KPPG	94.34	5.66

Tableau 5. Taux de reconnaissance en fonction de l'erreur des trois classifieurs sur NIST(85).

erreur	Max	$\leq 1.2 \%$	$\leq 0.4 \%$	$\leq 0.2 \%$
KPPV	96.83 %	93.5 %	81.59 %	73.73 %
SVM	97.90 %	—	—	—
KPPG	97.90 %	96.06 %	90.34 %	81.6 %

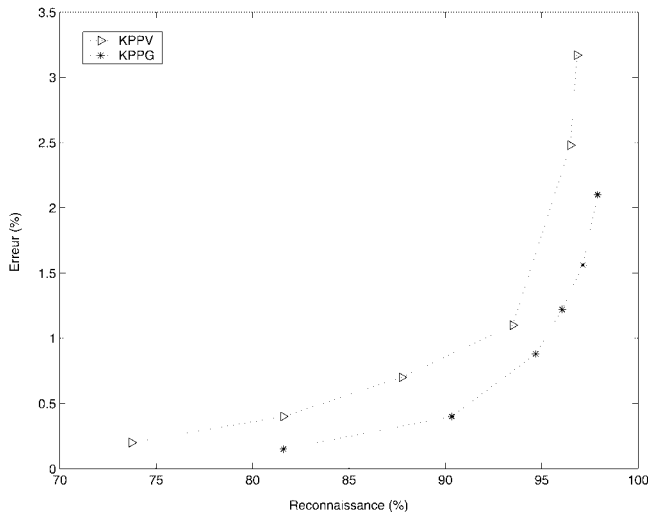


Figure 8. Erreur/Reconnaissance pour le KPPV et le KPPG sur la base NIST(85).

5. Conclusion

Dans ce papier, nous avons présenté une approche de conception d'un système multi-classifieurs basée sur la prise en compte de la topologie des données d'apprentissage. Cette conception est initiée par une phase préalable de clustering par cartes auto-organisatrices pour modéliser la topologie des données dans l'espace de représentation. Le modèle proposé montre performances au moins équivalentes aux meilleurs approches classiques, est incrémental en données et répond en cela au dilemme plasticité/stabilité contrairement à la plupart des modèles présentés dans la littérature. Ce modèle de cartes auto-organisatrices est utilisé ensuite pour piloter la conception d'un système d'apprentissage multi-classifieurs. Chaque neurone de la carte auto-organisée est associé à un ou plusieurs classifieurs dont le rôle est de performer au mieux sur le sous ensemble d'apprentissage définit pour ce neurone. Plusieurs stratégies de combinaison (coopération, compétition, fusion...) peuvent être mises en oeuvre à ce niveau. À ce stade de l'étude, seule une architecture à base de SVM a été utilisée. La conséquence directe et fondamentale à notre sens de cette stratégie concerne la définition de surfaces de décision fermées dans l'espace de représentation. Il en découle de réelles capacités de rejet bien adaptées aux problèmes de reconnaissance de formes, contrairement à la plupart des approches classiques. De plus, nous avons introduit un deuxième niveau de rejet à travers un coefficient de confiance s'affranchissant totalement du type de classifieurs utilisés, ce qui devrait permettre une amélioration de ces capacités de rejet. Les résultats préliminaires obtenus avec notre approche sont prometteurs. Les performances sont équivalentes aux autres classifieurs pour un taux d'erreur maximal, mais se révèlent supérieures quand des taux d'erreurs faibles ou fixés au préalable sont

recherchés. En effet, les résultats obtenus jusqu'à présent montrent que dans le cas où un taux de reconnaissance maximal est nécessaire, le système proposé est au moins aussi performant que les SVMs connus pour leur bonnes capacités. De plus, dans le cas où le taux d'erreur doit être faible, notre système obtient de bien meilleurs résultats. Notre méthode permet donc d'allier les performances en reconnaissance de certains classifieurs comme les SVMs et des capacités de rejet efficaces. Les propriétés de notre algorithme de clustering, et la liberté totale du choix de classifieurs, nous permettent de voir ce système comme une contribution pour la conception d'un système d'apprentissage fiable et incrémental. En effet, seul un classifieur de type SVM est associé aux neurones de la carte actuellement. Le fait de multiplier les classifieurs à ce niveau permettra probablement d'améliorer les résultats de manière significative. De la même manière, la mise en oeuvre du coefficient de confiance devrait contribuer à paramétrer efficacement le compromis erreur/rejet tout en intégrant des paramètres essentiels pour la modélisation du contexte de l'apprentissage et des contraintes de l'application.

Références

- [AOC*00] S. ADAM, J.M. OGIER, C. CARRIOU, R. MULLOT, J. LABICHE, J. GARDES, Symbol and character recognition: application to engineering drawings. *International Journal on Document Analysis and Recognition*, pages 89-101, 2000.
- [BB82] D.H. BALLARD, C.M. BROWN, *Computer Vision*. Prentice Hall, 1982.
- [DUI02] R.P.W. DUIN, The combining classifier: to train or not to train. In IEEE Computer Society, editor, *ICPR*, pages 765-771, Quebec, 2002.
- [FRI94] B. FRITZKE, Growing cell structures – A self-organizing network for unsupervised and supervised learning. *Neural Networks*, 7(9):1441-1460, 1994.
- [FRI95a] B. FRITZKE, Growing grid – a self organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letter*, (2):9-13, 1995.
- [FRI95b] B. FRITZKE, A Growing neural gas network learns topologies. In *Advances in Neural Information Processing Systems 7*, pages 625-632, 1995.
- [GA95] M. GUILLOT, R. AZOUZI, Improving on-line adaptation in neurocontrol using a combination of self-organizing map and multilayer feedforward network. *Intelligence Engineering Systems Through Artificial Neural Networks*, pages 915-922, 1995.
- [GR00] G. GIACINTO, F. ROLL, Dynamic classifier selection. In Springer, editor, *Proc. of the First Int. Workshop on Multiple Classifier Systems*, pages 177-189, juin 2000.
- [GS98] M. GORI, F. SCARSELLI, A Multilayer perceptron adequate for pattern recognition and verification. *IEEE Transactions on pattern analysis and machine intelligence*, 20:1121-1132, 1998.
- [HAY95] S. HAYKIN, *Neural Networks – A Comprehensive Foundations*. IEE Press, 1995.
- [HPG99] J.F. HEBERT, M. PARIZEAU, N. GHAZZALI, An hybrid architecture for active and incremental learning: The self-organizing perceptron (sop) network. In *IJCNN99*, Washington, DC, USA, July 1999.
- [JJNH91] R.A. JACOBS, M.J. JORDAN, S.J. NOWLAN and G.E. HINTON, Adaptive mixtures of local experts. *Neural Computation*, 3(1):79-87, 1991.

- [KOH82] T. KOHONEN, Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59-69, 1982.
- [KUN02] L.I. KUNCHEVA, A theoretical study on six classifier fusion strategies. In *IEEE Trans. Pattern Analysis and Machine Intelligence*, volume Pattern Recognition, Feb 2002.
- [LESG95] Y. LECOURTIER, A. ENNAJI, E. STOCKER, F. GILLES, Réseaux d'yprels et apprentissage incrémental. *Traitement du Signal*, 12(6):597-608, 1995.
- [LW03] K.W. LAU, Q.H. WU, Online training of support classifier. *Pattern Recognition*, 36:1913-1920, 2003.
- [MA92] P.M. MURPHY, D.W. AHA, UCI repository of machine learning databases. Machines-readable data repository, University of California, Department of Information and Computer Science, Irvine, CA, 1992.
- [MAC67] J.B. MACQUEEN, Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Symposium on Mathematics Statistics and Probability*, pages 281-297, 1967.
- [MAR93] T. MARTINEZ, Competitive Hebbian learning rule forms perfectly topology perservings maps. In *Proc. ICANN'93*, pages 427-434, Springer, 1993.
- [MS91] T. MARTINEZ, K. SCHULTEN, A "Meural-Gas" network learns topologies. In *Proc. ICANN*, volume I, pages 397-402, Amsterdam, Netherlands, 1991.
- [MS94] T. MARTINEZ, K. SCHULTEN, Topology representing networks. *Neural Networks*, 7(2), 1994.
- [PE04] Y. PRUDENT, A. ENNAJI, Clustering incrémental pour un apprentissage distribué. In *Conférence CAp 2004*, 2004.
- [RG02] F.ROLL, G. GIANCINTO, Design of multiple classifier systems. *Hybrid Methods in Pattern Recognition*, pages 199-226, April 2002.
- [RIB98] A. RIBERT, *Structuration Evolution de données : Application à la construction de classifieurs distribués*. PhD thesis, Université de Rouen, 1998.
- [TFCJW03] W. TING-FAN, L. CHIH-JEN, RUBY C.W., Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*, 5, 2004.
- [VAP95] V.N. VAPNIK, *The nature of statistical theory*. Springer, 1995.



Yann Prudent

Yann Prudent a reçu un DEA d'Informatique à l'Université de Montpellier 2 en 2003. Il poursuit actuellement une thèse à l'Université de Rouen sur les problèmes d'apprentissage incrémental et d'analyse de données.



Abdellatif Ennaji

Abdellatif Ennaji est Maître de Conférences à l'Université de Rouen depuis 1993. Il a soutenu un doctorat de l'Université de Rouen en 1993 sur le thème de l'apprentissage et les réseaux de neurones dans des applications de reconnaissance de formes. Ses activités de recherche actuelles portent essentiellement sur les problèmes d'apprentissage, d'analyse de données, et plus particulièrement l'apprentissage incrémental en données. Ces travaux sont principalement appliqués en reconnaissance de formes et en reconnaissance de textes arabes